# Task 5 — Capture and Analyze Network Traffic Using Wireshark

**Date:** 11 August 2025  
**Submitted by:** PRADEEP S

## Objective

To capture live network traffic using Wireshark, analyze the captured packets to identify common protocols such as DNS, HTTP, TCP, and ICMP, and understand their roles in network communication. This helps develop practical skills in packet analysis, protocol filtering, and interpreting network interactions for troubleshooting and cybersecurity purposes.

## Tools Used

* **Wireshark** — For capturing and analyzing network packets.
* **Command Prompt (ping command)** — To generate ICMP traffic for analysis.

## Steps Performed

1. Installed and launched Wireshark on the system.
2. Selected the active network interface (Wi‑Fi) to capture live traffic.
3. Started packet capture in Wireshark.
4. Opened Command Prompt and ran:

* ping google.com
* to generate ICMP (ping) traffic.

1. Opened a web browser and visited a website to create HTTP/DNS traffic.
2. Stopped the capture after about one minute.
3. Applied protocol filters in Wireshark (e.g., dns, tcp, http, icmp) to analyze specific traffic.
4. Identified different protocols and noted important packet details.
5. Saved the capture file as task5\_capture.pcapng.

## Results — First 10 packets (sample)

These are representative packets observed at the start of the capture:

1. 0.000000 Your\_IP → DNS\_Server DNS Standard query A google.com
2. 0.050000 DNS\_Server → Your\_IP DNS Standard query response A 142.250.x.x
3. 0.070000 Your\_IP → 142.250.x.x TCP SYN
4. 0.080000 142.250.x.x → Your\_IP TCP SYN, ACK
5. 0.090000 Your\_IP → 142.250.x.x TCP ACK
6. 0.100000 Your\_IP → 142.250.x.x HTTP GET /
7. 0.120000 142.250.x.x → Your\_IP HTTP 200 OK

(Note: timestamps and IPs are sample values from the capture.)

## Protocols Identified

* **DNS** — Resolves domain names to IP addresses (typically UDP port 53).
* **TCP** — Reliable transport protocol used by HTTP/HTTPS (three‑way handshake seen).
* **HTTP / HTTPS (TLS)** — Web traffic; TLS handshakes observed for encrypted HTTPS on port 443.
* **ICMP** — Echo (ping) requests and replies used for connectivity checks.
* **UDP** — Connectionless traffic from some background services or apps.
* **ARP / ICMPv6 / IGMP** — Observed as local network and multicast management traffic in the capture.

## Traffic Flow (Simple Explanation)

1. **DNS Lookup:** The computer first asks a DNS server to find the IP address of websites like google.com or assets.msn.com.
2. **TCP Connection:** The computer and server perform a TCP three‑way handshake (SYN, SYN‑ACK, ACK) to establish a reliable connection.
3. **Ping Test (ICMP):** The computer sends ICMP ping requests to check if the server is reachable and receives replies.
4. **Other Activity (UDP/background):** Some apps or system services continuously send UDP packets and DNS requests in the background.
5. **HTTP/HTTPS Connection:** After the TCP handshake, the browser sends an HTTP or HTTPS request; the server responds with the requested data (HTTPS is encrypted via TLS).

## Observations

* DNS queries occur before web connections to resolve hostnames.
* A TCP three‑way handshake precedes HTTP/HTTPS data transfer.
* ICMP ping confirms network reachability with replies and round‑trip times.
* Presence of UDP and repeated DNS queries suggests background application or OS activity.
* TLSv1.2 (or similar) handshakes are seen for secure web connections — packet contents are encrypted.

## Outcome

Completed hands‑on packet capture and analysis using Wireshark. Identified multiple protocols, practiced filtering and packet inspection, and gained insights into how name resolution, connection establishment, and data transfer occur on the network.

**Files associated with this task:**

* task5\_capture.pcapng (packet capture)
* README.md (this file)
* screenshot.png (Wireshark screenshot, optional)